

ggplot2

Ana Ruiz-Garcia

September 12, 2022

Library

For this tutorial we will use the tidyverse package, which includes, among others, ggplot2 and dplyr libraries.

```
#install.packages("mrggsave",repo="https://mpn.metworx.com/snapshots/stable/2022-06-15")
library(tidyverse)
```

```
## -- Attaching packages ----- tidyverse 1.3.2 --
## v ggplot2 3.3.6      v purrr   0.3.4
## v tibble  3.1.7      v dplyr   1.0.9
## v tidyr   1.2.0      v stringr 1.4.0
## v readr   2.1.2      v forcats 0.5.1
## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(gridExtra)
```

```
##
## Attaching package: 'gridExtra'
##
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
library(plyr)
```

```
## -----
## You have loaded plyr after dplyr - this is likely to cause problems.
## If you need functions from both plyr and dplyr, please load plyr first, then dplyr:
## library(plyr); library(dplyr)
## -----
##
## Attaching package: 'plyr'
##
## The following objects are masked from 'package:dplyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize
```

```
##
## The following object is masked from 'package:purrr':
##
## compact
```

```
library(mrggsave)

figDir <- file.path("Figures")
options(mrggsave.dir = figDir, mrg.script = "ggplot.R")
```

#Understanding the Data

Let's open the simulated nonmem output and add names to the different variables:

```
names<-c("REP", "ID", "ROUTE","AMT", "TIME", "DAY", "TAD", "DV" ,
         "FOOD", "SEX" , "BWT", "DOSE","EVID" , "MDV" , "CMT", "CMAX", "TMAX", "START")

d <- read.table("simu34",header = FALSE, skip = 0)

names(d)<-names
head(d)
```

```
##      REP ID ROUTE    AMT TIME DAY TAD      DV FOOD SEX      BWT DOSE EVID MDV CMT
## 1    1  1  2    20000  0.0  1  0.0  0.0000    1  0 80.2063   20    1  1  1
## 2    1  1  2      0  0.1  1  0.1  2.1381    1  0 80.2063   20    0  0  2
## 3    1  1  2      0  1.0  1  1.0  3.9089    1  0 80.2063   20    0  0  2
## 4    1  1  2      0  2.0  1  2.0  3.9453    1  0 80.2063   20    0  0  2
## 5    1  1  2      0  4.0  1  4.0  3.9089    1  0 80.2063   20    0  0  2
## 6    1  1  2      0  6.0  1  6.0  3.6900    1  0 80.2063   20    0  0  2
##      CMAX TMAX START
## 1  0.0000    0    0
## 2  0.0000    0    0
## 3  7.7691    1    0
## 4 42.6622    2    0
## 5 49.6828    4    0
## 6 49.6828    4    0
```

For nicer plots we will manipulate the dataset to add labels to some of the variables: FOOD and SEX.

```
d.label<-d%>%
  mutate(AMT= ifelse(AMT==0,NA,AMT),FOODlabel=ifelse(FOOD==0,"Fasted","Fed"),
         sexlabel=ifelse(SEX==0,"Male","Female"),
         routelabel=ifelse(ROUTE==2,"Oral","IV"),
         EXPDV=ifelse(DV==0,0,exp(DV)))%>%
  filter(MDV==0)
```

We will save the resulting dataset as a csv file.

```
write.csv(d.label,"PKdata_plots.csv",quote=F,na=".",row.names = F)
```

Creating Plots

Ggplot is based in the grammar of graphics, you can build every graph from the same components: a dataset, a coordinate system, and geoms- visual marks that represent data points. A cheatsheet for your reference can be found at <https://www.rstudio.com/wp-content/uploads/2015/03/ggplot2-cheatsheet.pdf>

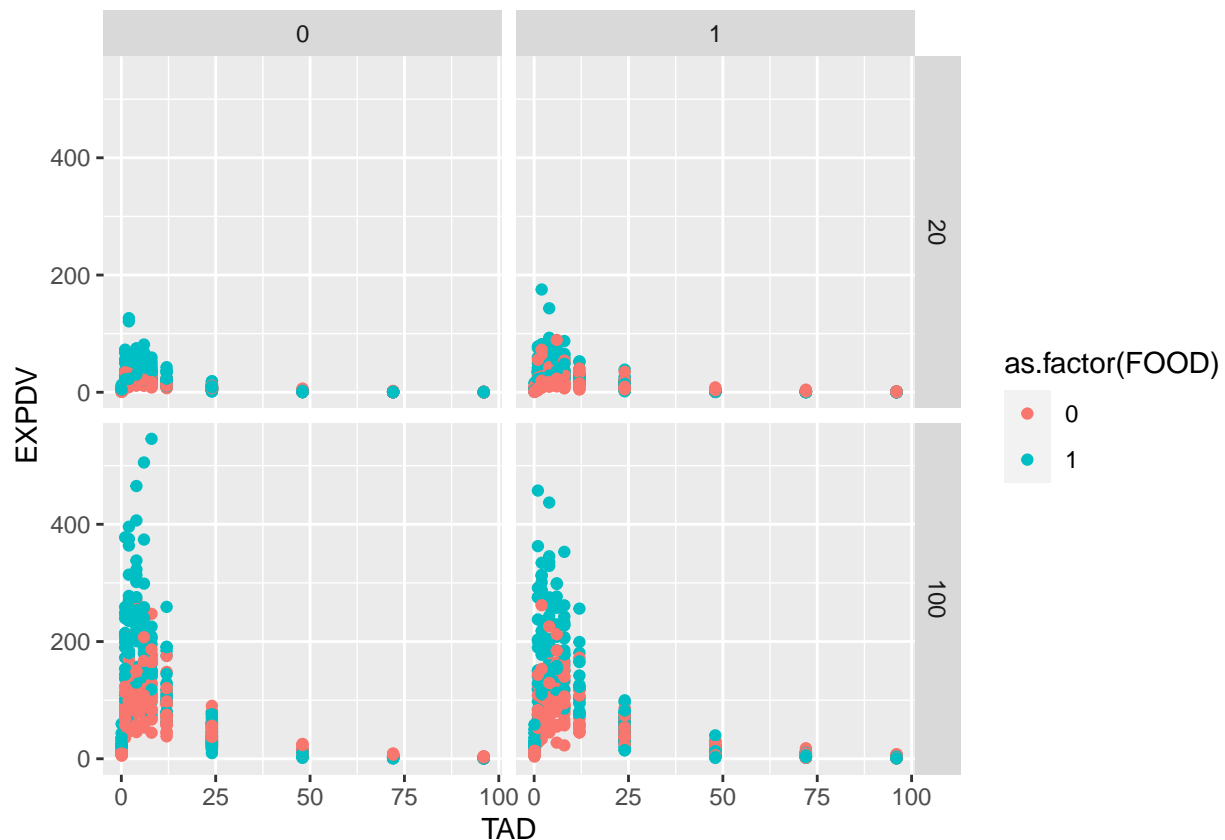
Understanding your dataset is necessary to obtain the desired plots. This simulated dataset contains single dose data (DAYS 1-5 and 18-22) and steady-state data (DAYS 11 and 28) after oral administration with [FOOD=1] and without food [FOOD=0] at 2 dose levels (20 and 100 mg). The half-life of the drug is approximately 22 hrs. IV administration after single dose is also included (ROUTE=1).

Let's create single dose datasets for oral and IV administrations and a steady-state dataset using dplyr:

```
sd.oral<-d.label %>% filter(DAY>=1 & DAY<=5 & ROUTE==2 | DAY>=18 & DAY<=22 & ROUTE==2 )
sd.iv<-d.label %>% filter(DAY>=1 & DAY<=5 & ROUTE==1 | DAY>=18 & DAY<=22 & ROUTE==1 )
ss<-d.label %>% filter(DAY==11 | DAY==28 )
```

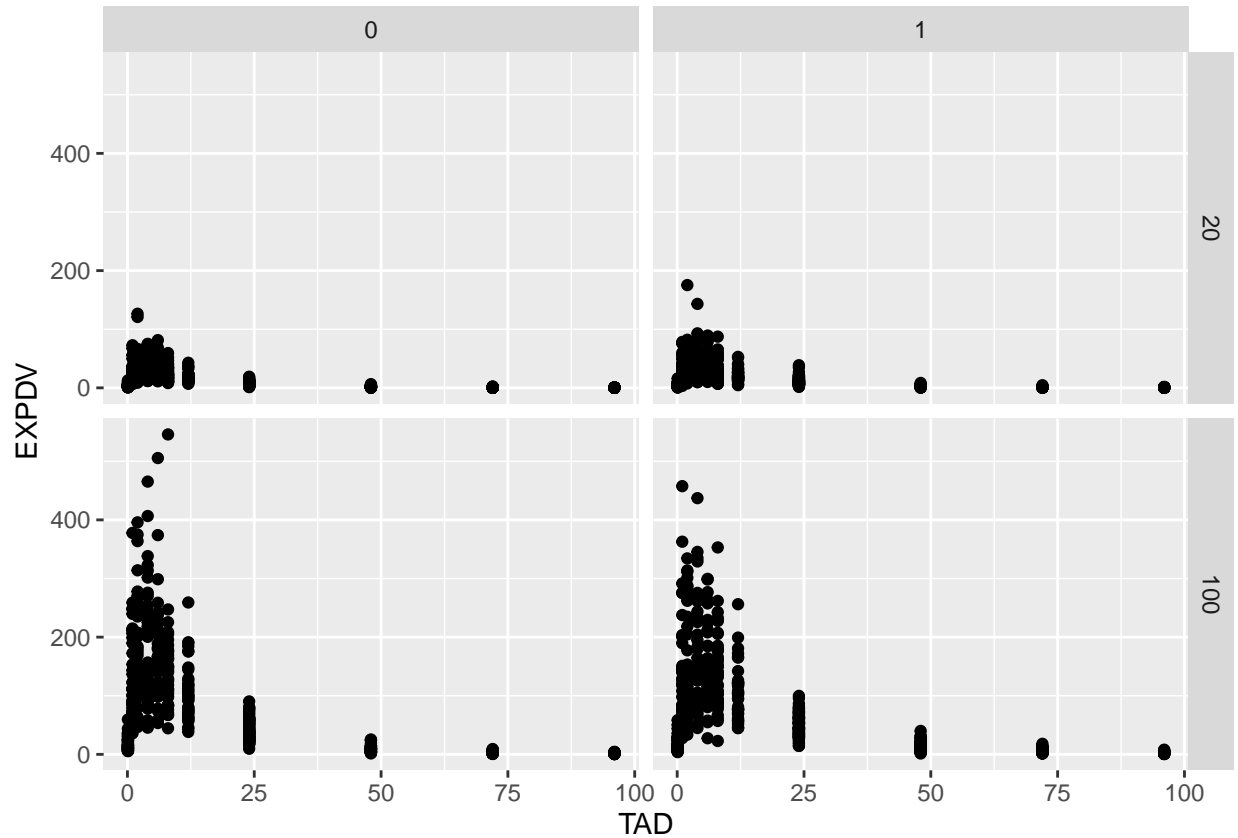
Now let's create our first plot with oral single dose data at 2 dose levels

```
sdplot<-ggplot(sd.oral,aes(x=TAD, y=EXPDV, colour=as.factor(FOOD))) +
  geom_point()+
  facet_grid(DOSE~SEX)
sdplot
```



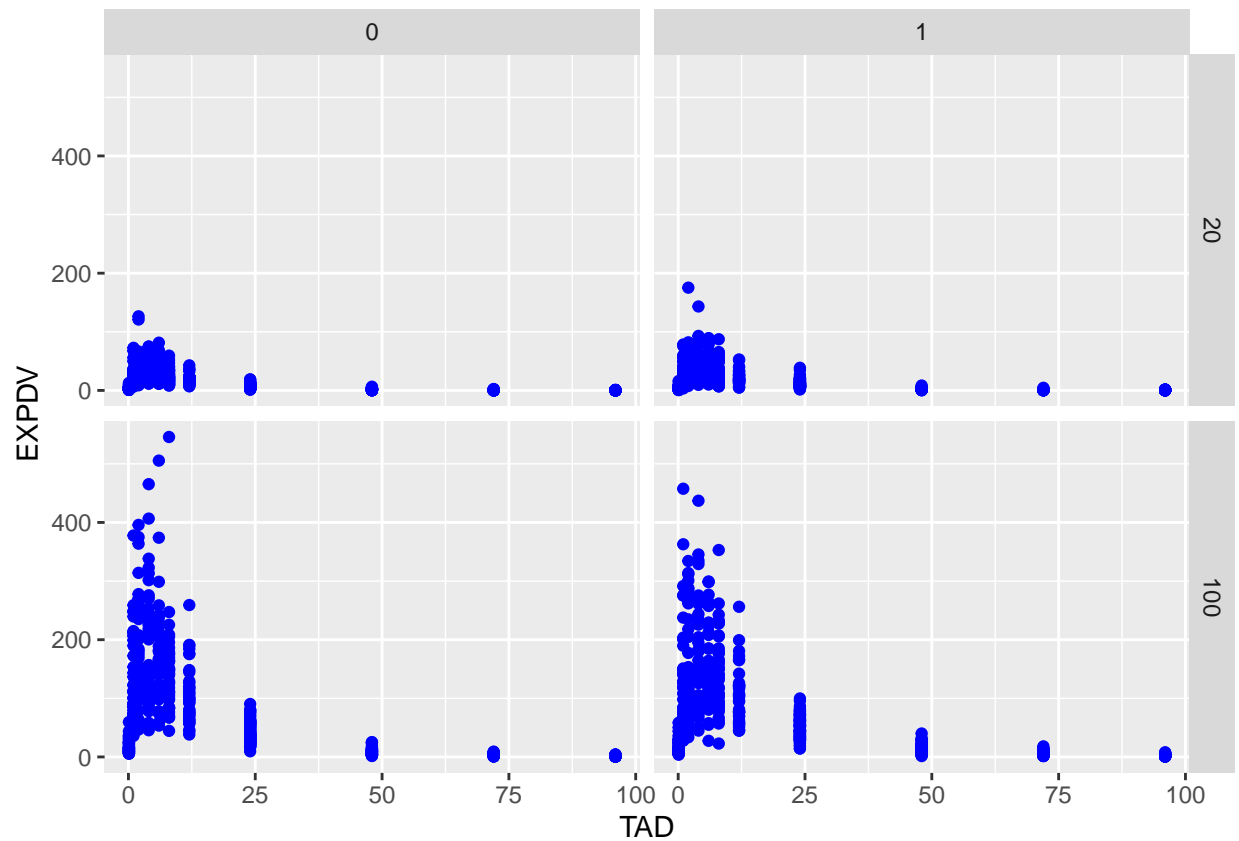
What happened if color is outside aes?

```
sdplot<-ggplot(sd.oral,aes(x=TAD, y=EXPDV), colour=as.factor(FOOD)) +  
  geom_point()+  
  facet_grid(DOSE~SEX)  
  
sdplot
```



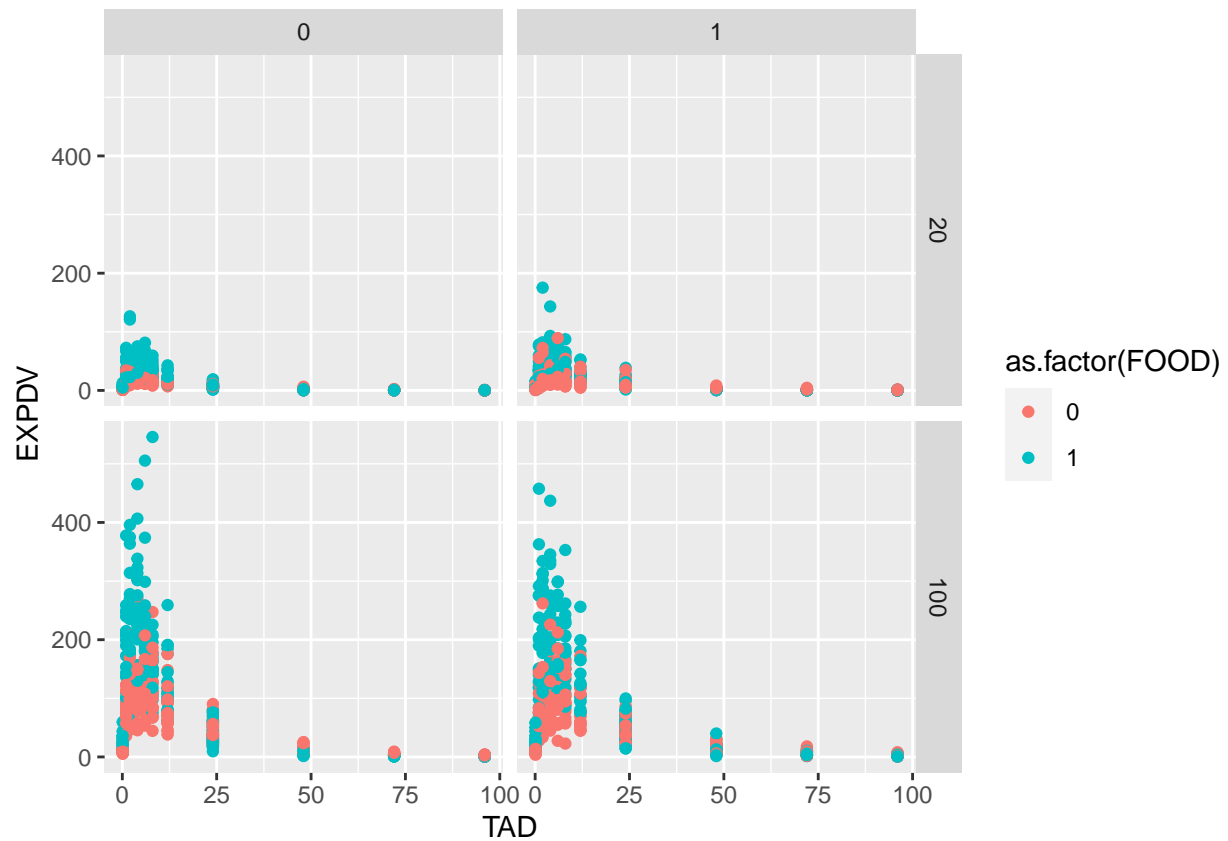
ggplot2 only understand data variables under aes. Alternatively you can define colour outside aes as follows:

```
sdplot<-ggplot(sd.oral,aes(x=TAD, y=EXPDV)) +  
  geom_point(colour="blue")+  
  facet_grid(DOSE~SEX)  
  
sdplot
```



The data variable could be used to specify color inside aes:

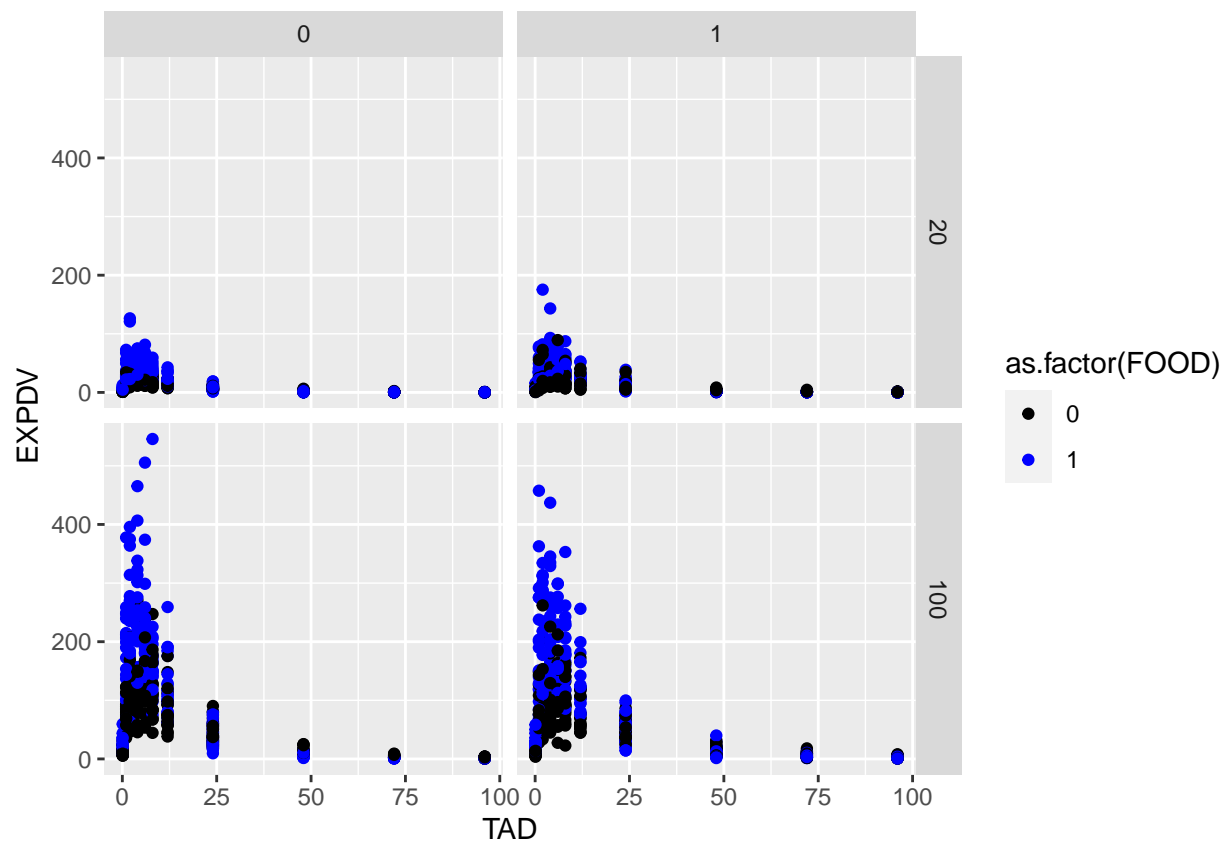
```
sdplot<-ggplot(sd.oral,aes(x=TAD, y=EXPDV)) +  
  geom_point( aes(colour=as.factor(FOOD)))+  
  facet_grid(DOSE~SEX)  
  
sdplot
```



We can modify the colour selected when specifying data variable using `scale_colour_manual`

```
sdplot<-ggplot(sd.oral,aes(x=TAD, y=EXPDV)) +
  geom_point(aes(colour=as.factor(FOOD)))+
  scale_colour_manual(values = c("black","blue"))+
  facet_grid(DOSE~SEX)

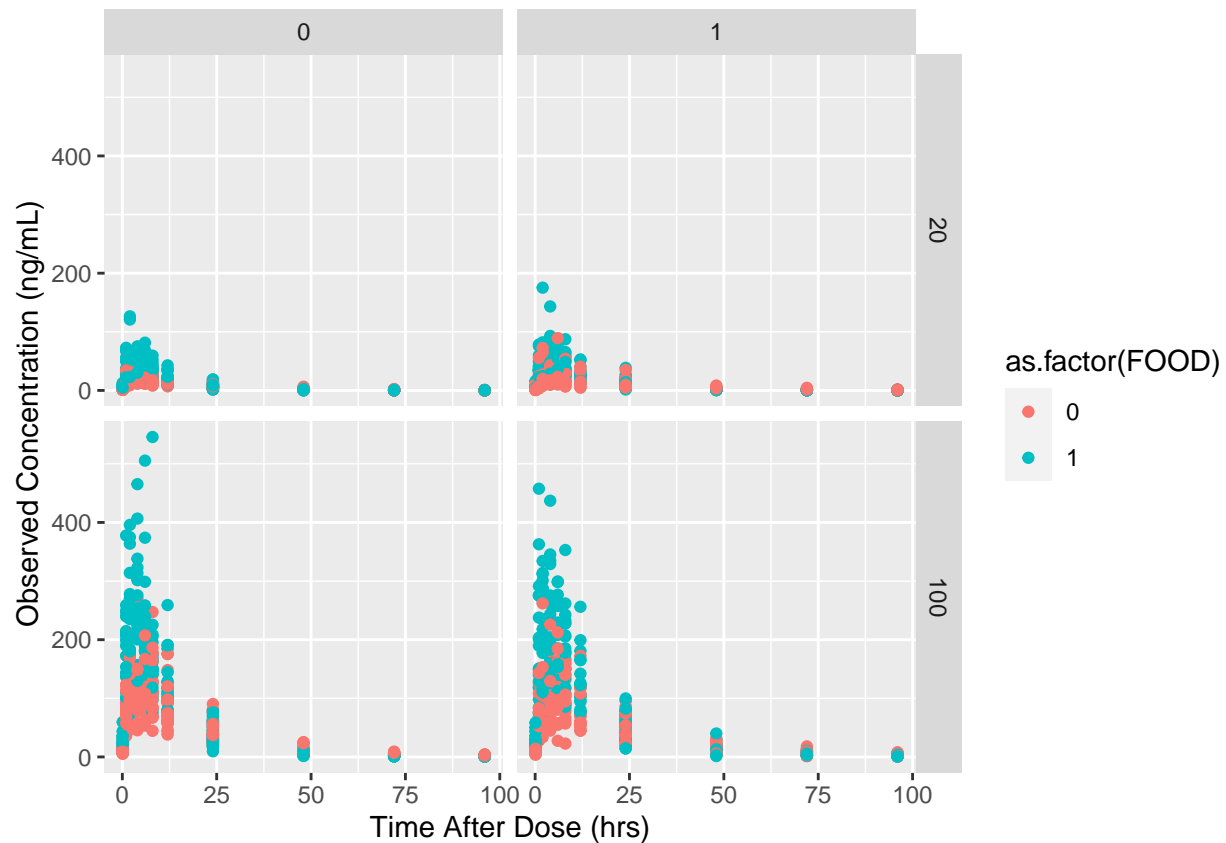
sdplot
```



We can customize the axis and the legends that appear in the

```
sdplot2<-ggplot(sd.oral,aes(x=TAD, y=EXPDV, colour=as.factor(FOOD))) +
  geom_point()+
  facet_grid(DOSE~SEX)+
  xlab("Time After Dose (hrs)") +
  ylab("Observed Concentration (ng/mL)")

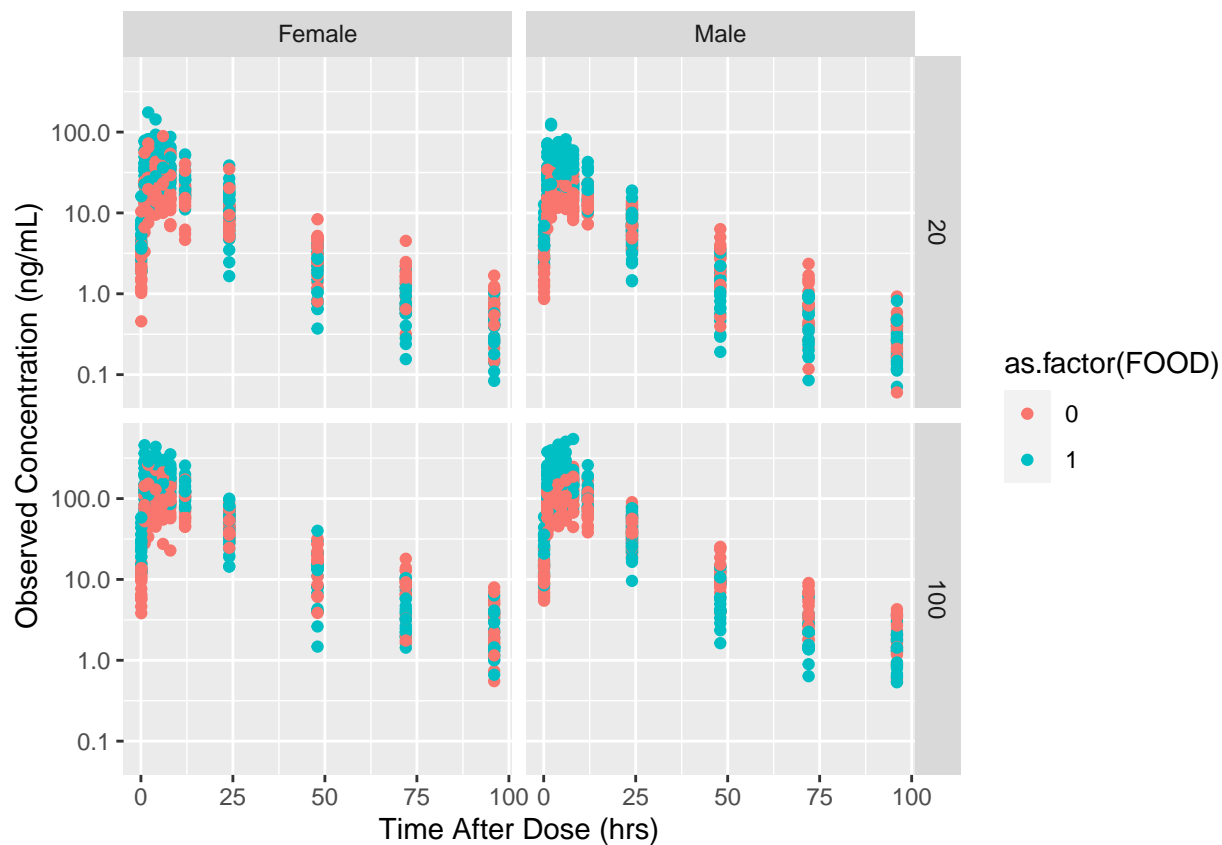
sdplot2
```



If we use sexlabel instead of the variable SEX and we add logY axis:

```
sdplot3<-ggplot(sd.oral,aes(x=TAD, y=EXPDV, colour=as.factor(FOOD))) +
  geom_point()+
  facet_grid(DOSE~sexlabel)+
  xlab("Time After Dose (hrs)") +
  ylab("Observed Concentration (ng/mL)")+
  scale_y_log10()

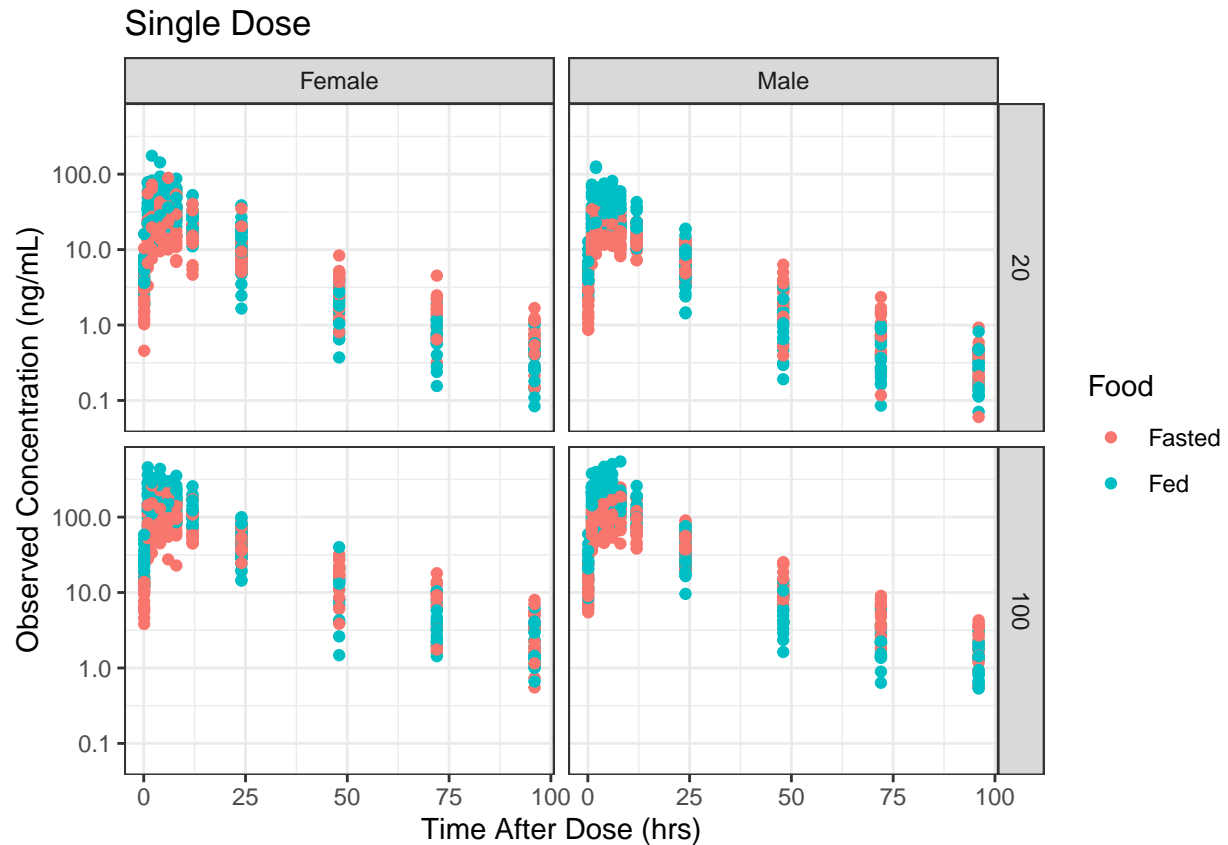
sdplot3
```

We can remove the gray background, add main title, and customize the legend:

```
sdplot4<-ggplot(sd.oral,aes(x=TAD, y=EXPDV, colour=as.factor(FOOD))) +
  geom_point()+
  facet_grid(DOSE~sexlabel)+
  theme_bw() +
  xlab("Time After Dose (hrs)") +
  ylab("Observed Concentration (ng/mL)")+
  scale_y_log10()+
  ggtitle("Single Dose")+
  scale_colour_discrete(name="Food", # Legend label, use darker colors
                        labels=c("Fasted", "Fed"))

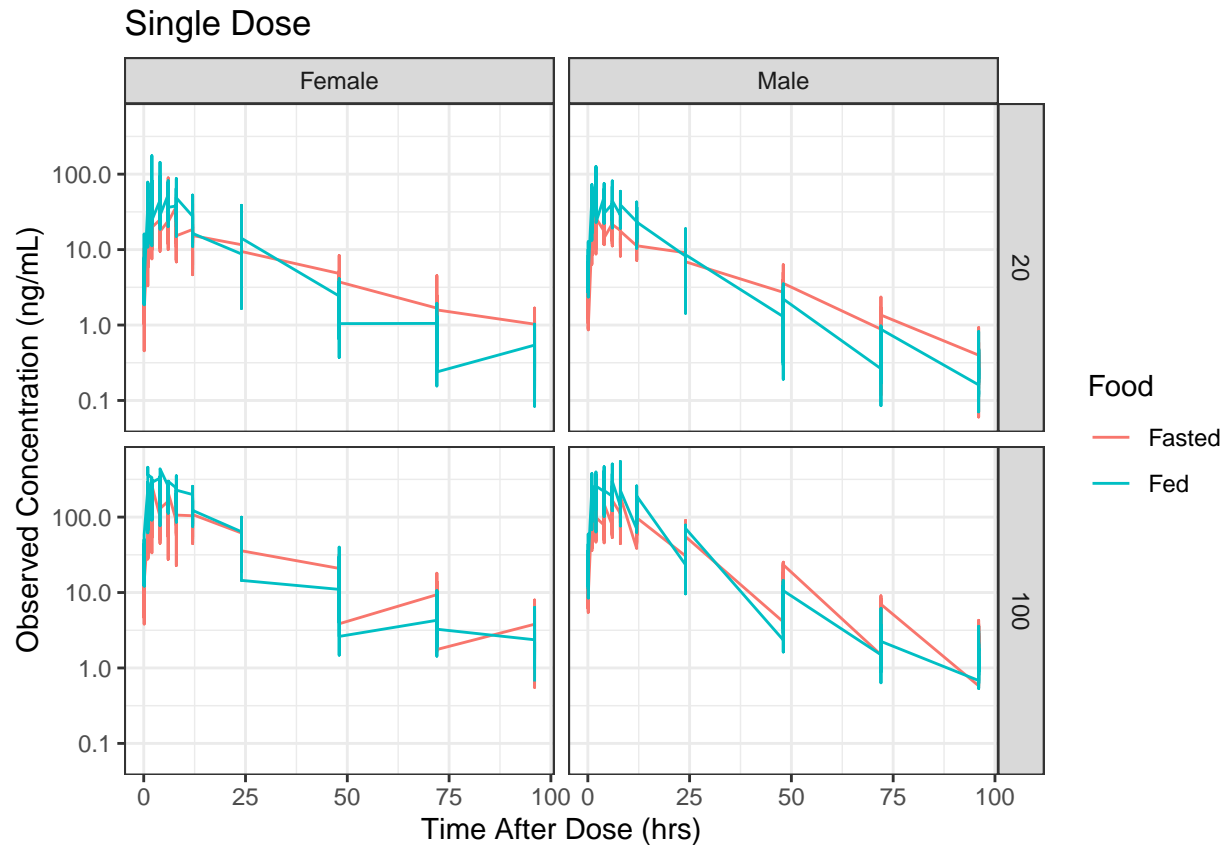
sdplot4
```



Instead of points (`geom_point()`) we can use lines:

```
sdplot5<-ggplot(sd.oral,aes(x=TAD, y=EXPDV, colour=as.factor(FOOD))) +
  geom_line()+
  facet_grid(DOSE~sexlabel)+
  theme_bw() +
  xlab("Time After Dose (hrs)") +
  ylab("Observed Concentration (ng/mL)")+
  scale_y_log10()+
  ggtitle("Single Dose")+
  scale_colour_discrete(name="Food", # Legend label, use darker colors
                        labels=c("Fasted", "Fed"))

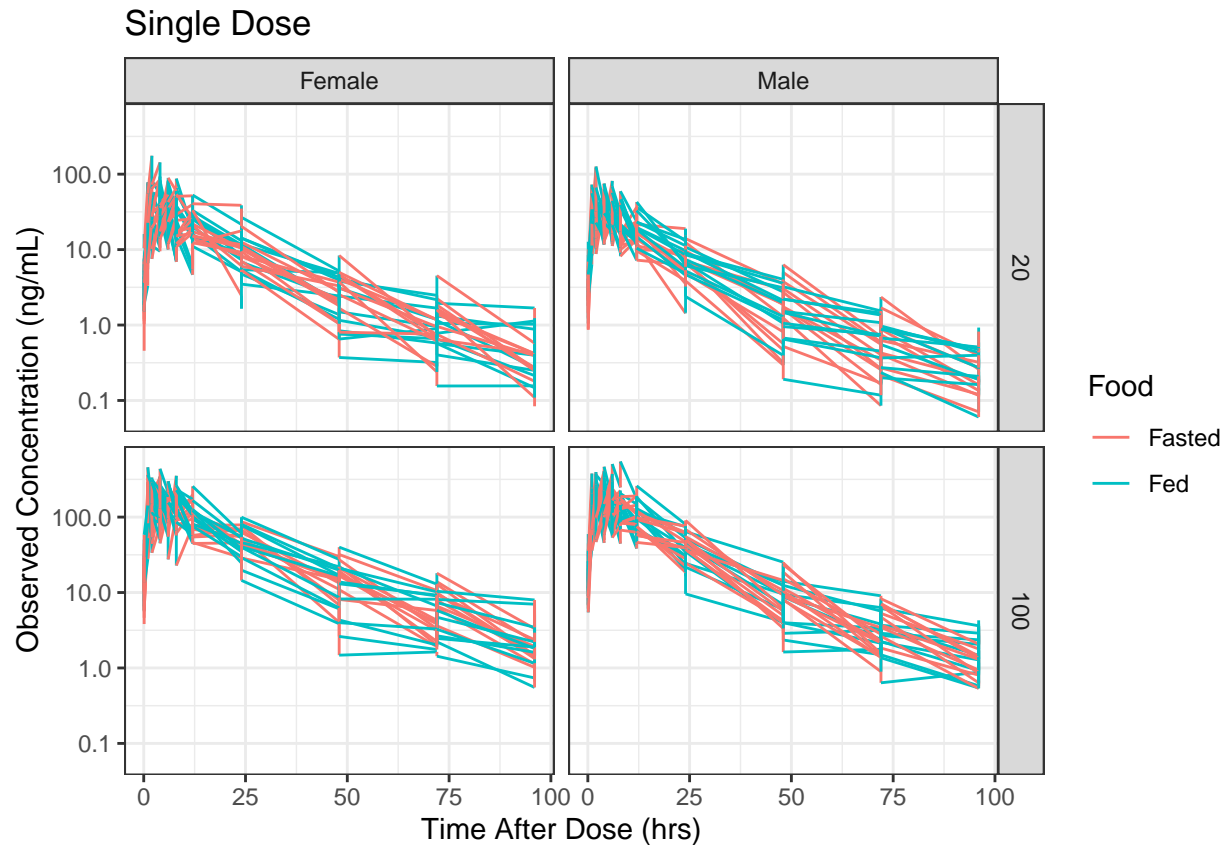
sdplot5
```



In this case a line representing the pool of observations is drawn, If we want to see individual trajectories we should add group:

```
sdplot6<-ggplot(sd.oral,aes(x=TAD, y=EXPDV, group=ID, colour=as.factor(FOOD))) +
  geom_line()+
  facet_grid(DOSE~sexlabel)+
  theme_bw() +
  xlab("Time After Dose (hrs)") +
  ylab("Observed Concentration (ng/mL)")+
  scale_y_log10()+
  ggtitle("Single Dose")+
  scale_colour_discrete(name="Food", # Legend label, use darker colors
    labels=c("Fasted", "Fed"))

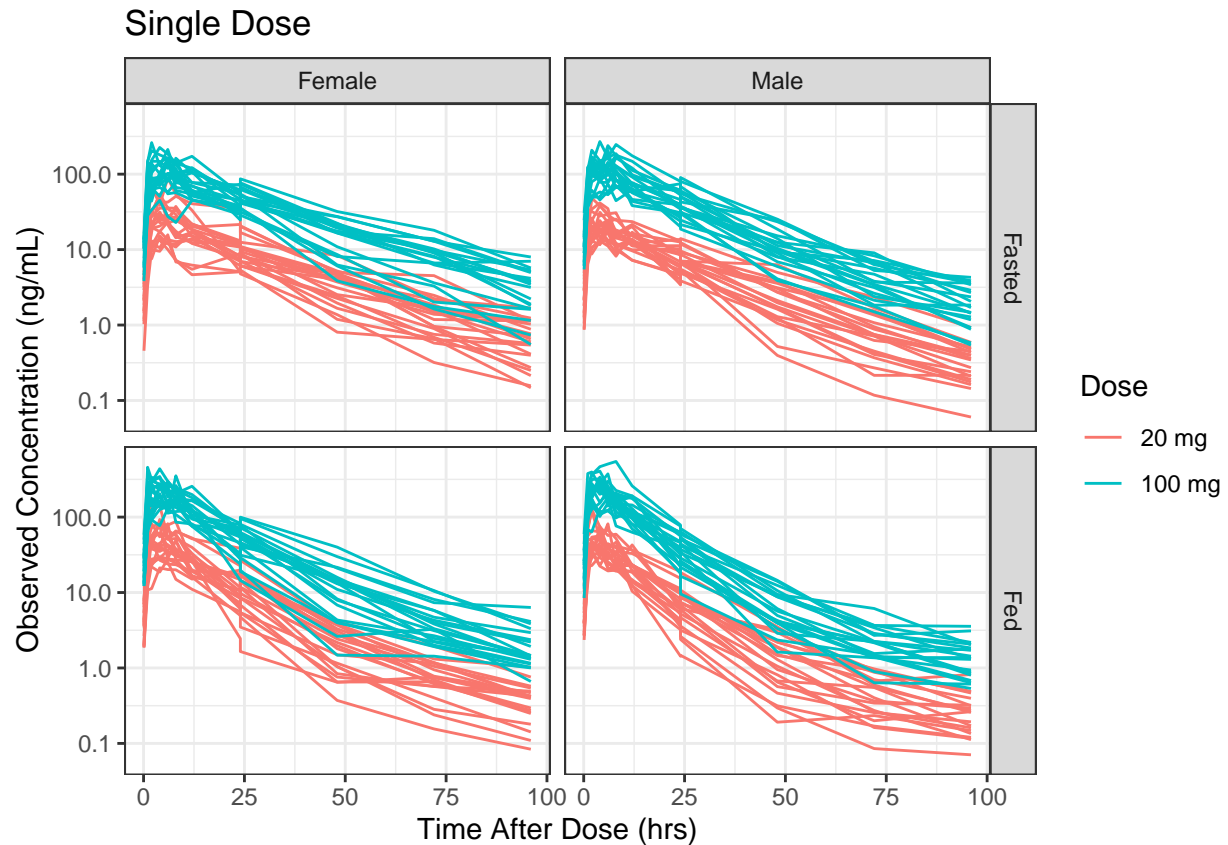
sdplot6
```



However, as the same ID received FODD=0 and FOOD=1, the lines don't represent individual occasion trajectories, some additional tweaking is required:

```
sdplot7<-ggplot(sd.oral,aes(x=TAD, y=EXPDV, group=ID, colour=as.factor(DOSE))) +
  geom_line()+
  facet_grid(FOODlabel~sexlabel)+
  theme_bw() +
  xlab("Time After Dose (hrs)") +
  ylab("Observed Concentration (ng/mL)")+
  scale_y_log10()+
  ggtitle("Single Dose")+
  scale_colour_discrete(name="Dose", # Legend label, use darker colors
                        labels=c("20 mg", "100 mg"))

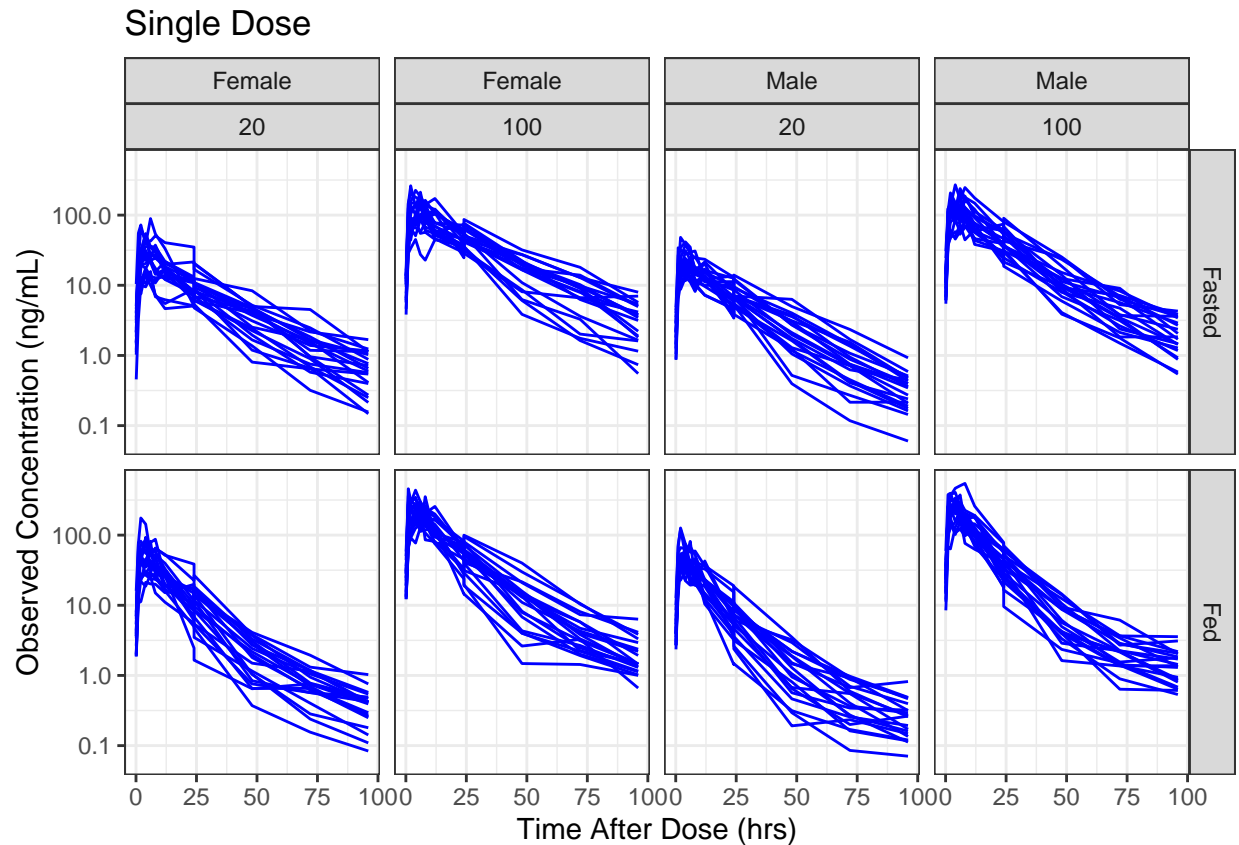
sdplot7
```



An alternative could be:

```
sdplot7b<-ggplot(sd.oral,aes(x=TAD, y=EXPDV, group=ID)) +
  geom_line(colour="blue")+
  facet_grid(FOODlabel~sexlabel+DOSE)+
  theme_bw() +
  xlab("Time After Dose (hrs)") +
  ylab("Observed Concentration (ng/mL)")+
  scale_y_log10()+
  ggtitle("Single Dose")+
  scale_colour_discrete(name="Dose", # Legend label, use darker colors
                        labels=c("20 mg", "100 mg"))

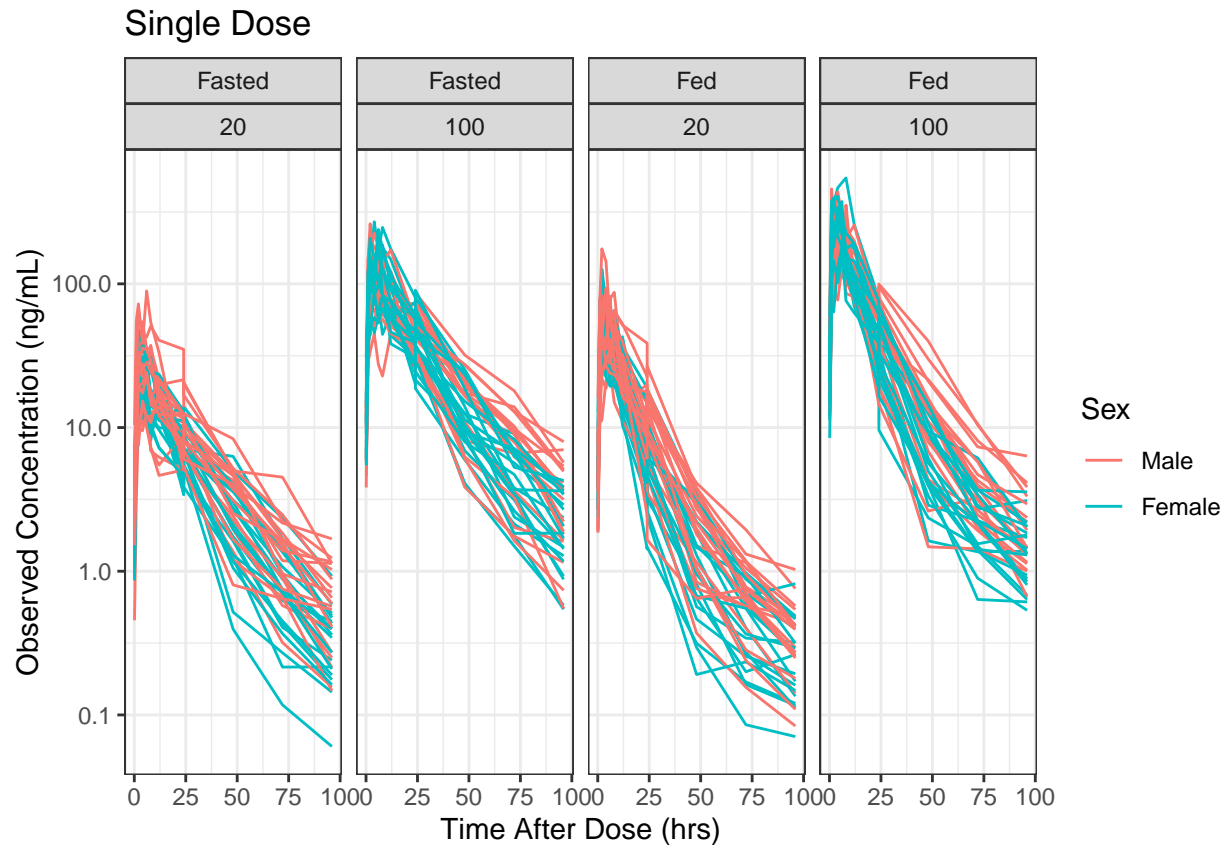
sdplot7b
```



or:

```
sdplot7c<-ggplot(sd.oral,aes(x=TAD, y=EXPDV, group=ID,colour=sexlabel)) +
  geom_line()+
  facet_grid(.~FOODlabel+DOSE)+
  theme_bw() +
  xlab("Time After Dose (hrs)") +
  ylab("Observed Concentration (ng/mL)")+
  scale_y_log10()+
  ggtitle("Single Dose")+
  scale_colour_discrete(name="Sex", # Legend label, use darker colors
                        labels=c("Male", "Female"))

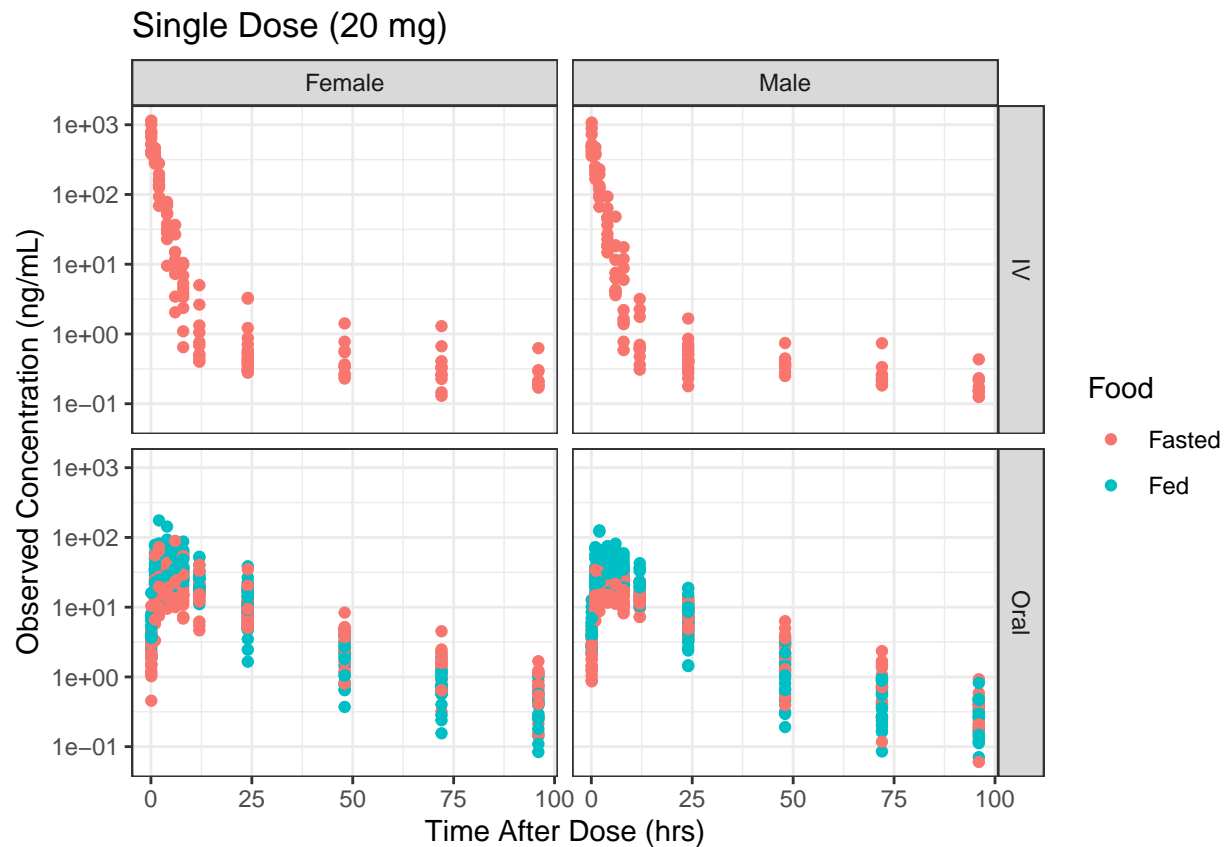
sdplot7c
```



We can now do a similar exercise with IV data and steady-state datasets, or we can use the original dataset and try to subset data within the plot:

```
p<-ggplot(d.label%>%filter(DAY>=1 & DAY<=5 & DOSE==20| DAY>=18 & DAY<=22 & DOSE==20),aes(x=TAD, y=EXPDI))
  geom_point()+
  facet_grid(routelabel~sexlabel)+
  theme_bw() +
  xlab("Time After Dose (hrs)") +
  ylab("Observed Concentration (ng/mL)") +
  scale_y_log10()+
  ggtitle("Single Dose (20 mg)") +
  scale_colour_discrete(name="Food", # Legend label, use darker colors
    labels=c("Fasted", "Fed"))
```

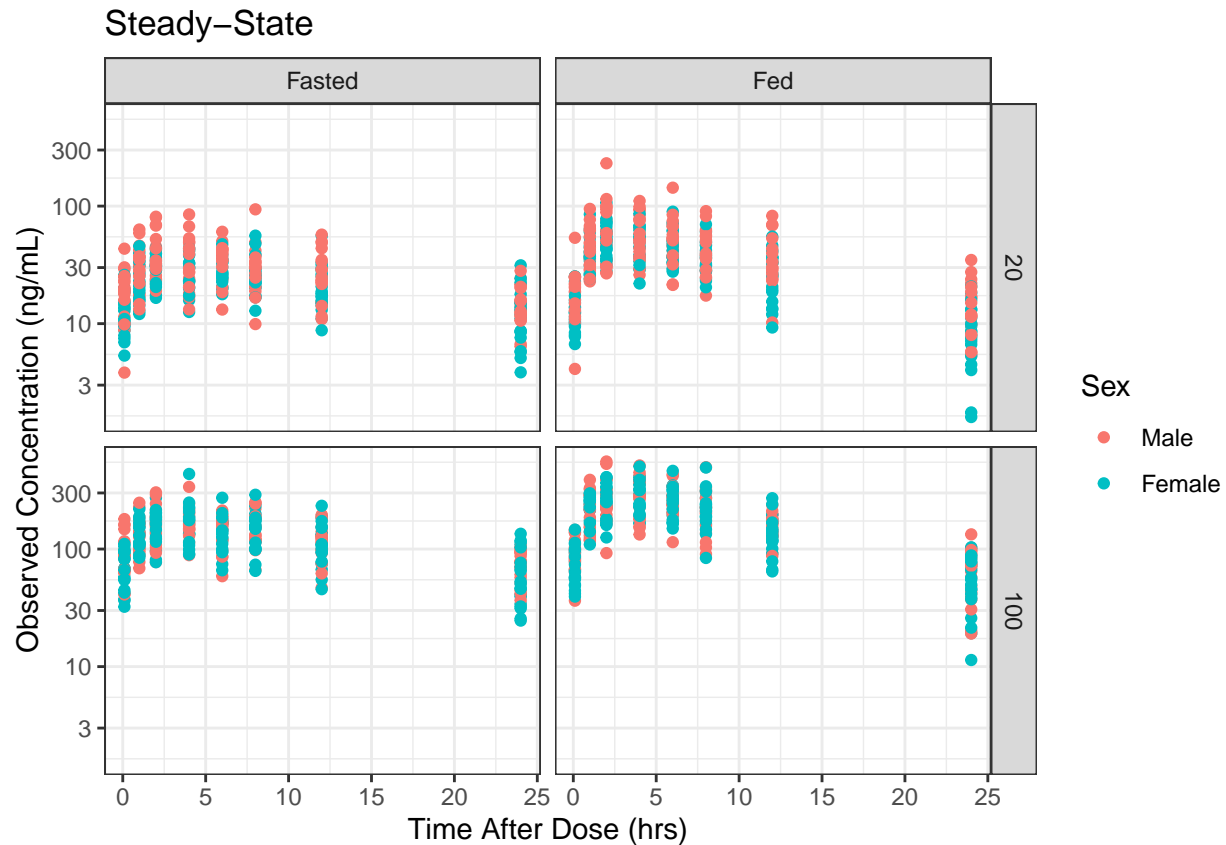
p



For steady-State data

```
p2<-ggplot(d.label%>%filter(DAY==11| DAY==28 ),aes(x=TAD, y=EXPDV, group=(ID),colour=as.factor(sexlabel))
  geom_point()+
  facet_grid(DOSE~FOODlabel)+
  theme_bw() +
  xlab("Time After Dose (hrs)") +
  ylab("Observed Concentration (ng/mL)")+
  scale_y_log10()+
  ggtitle("Steady-State")+
  scale_colour_discrete(name="Sex", # Legend label, use darker colors
    labels=c("Male", "Female"))
```

p2

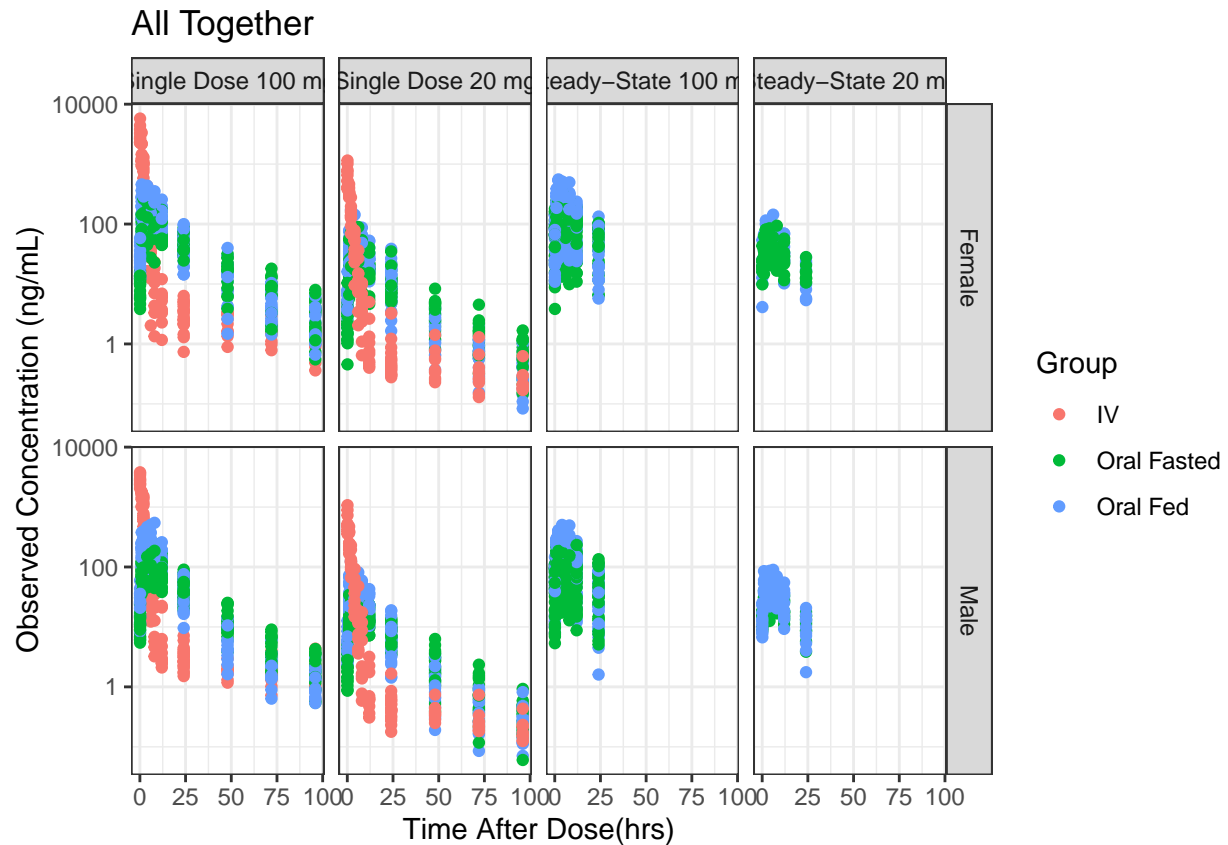


We can plot everything in one figure playing with shapes, color and facet. Adding a new variable for stratification may help to obtain the look we may want.

```
d.all<-d.label %>%
  mutate(label2=ifelse(DAY>=1 & DAY<=5 & DOSE==20| DAY>=18 & DAY<=22 & DOSE==20,"Single Dose 20 mg",
    ifelse(DAY==1 & DOSE==20| DAY==28 & DOSE==20,"Steady-State 20 mg",
      ifelse(DAY>=1 & DAY<=5 & DOSE==100| DAY>=18 & DAY<=22 & DOSE==100,"Single Dose 100 mg", "Steady-State 100 mg"),
    label3=ifelse(routelabel=="Oral" & FOODlabel=="Fasted", "Oral Fasted",
      ifelse(routelabel=="Oral" & FOODlabel=="Fed", "Oral Fed", "IV"))))

p3<-ggplot(d.all,aes(x=TAD, y=EXPDV, colour=as.factor(label3))) +
  geom_point()+
  facet_grid(sexlabel~label2)+
  theme_bw() +
  xlab("Time After Dose(hrs)") +
  ylab("Observed Concentration (ng/mL)")+
  scale_y_log10()+
  ggtitle("All Together")+
  scale_colour_discrete(name="Group", # Legend label, use darker colors
    labels=c("IV", "Oral Fasted", "Oral Fed"))

p3
```



```
## Create an Average curve
```

Lets summarize the data first by the variables we would like to present the data: FOOD, SEX, ROUTE, DOSE, SD vs SS and TAD:

```
df.ss <- d.all %>%
  group_by(label2, label3, FOODlabel, sexlabel, TAD) %>%
  dplyr::summarize( N = length(EXPDV), CONC = mean(EXPDV), med = median(EXPDV), sd = sd(EXPDV))
```

```
## 'summarise()' has grouped output by 'label2', 'label3', 'FOODlabel',
## 'sexlabel'. You can override using the '.groups' argument.
```

```
head(df.ss)
```

```
## # A tibble: 6 x 9
## # Groups:   label2, label3, FOODlabel, sexlabel [1]
##   label2      label3 FOODlabel sexlabel  TAD    N  CONC   med    sd
##   <chr>      <chr>   <chr>    <chr>  <dbl> <int> <dbl> <dbl> <dbl>
## 1 Single Dose 100 mg IV      Fasted   Female    0.1    10 3227. 2742. 1135.
## 2 Single Dose 100 mg IV      Fasted   Female    1      10 1567. 1190.  773.
## 3 Single Dose 100 mg IV      Fasted   Female    2      10  775.  797.  334.
## 4 Single Dose 100 mg IV      Fasted   Female    4      10  163.  137.  114.
## 5 Single Dose 100 mg IV      Fasted   Female    6      10  45.3  34.1  50.1
## 6 Single Dose 100 mg IV      Fasted   Female    8      10  18.8  12.1  26.7
```

Using our ggplot2 knowledge now we can plot:

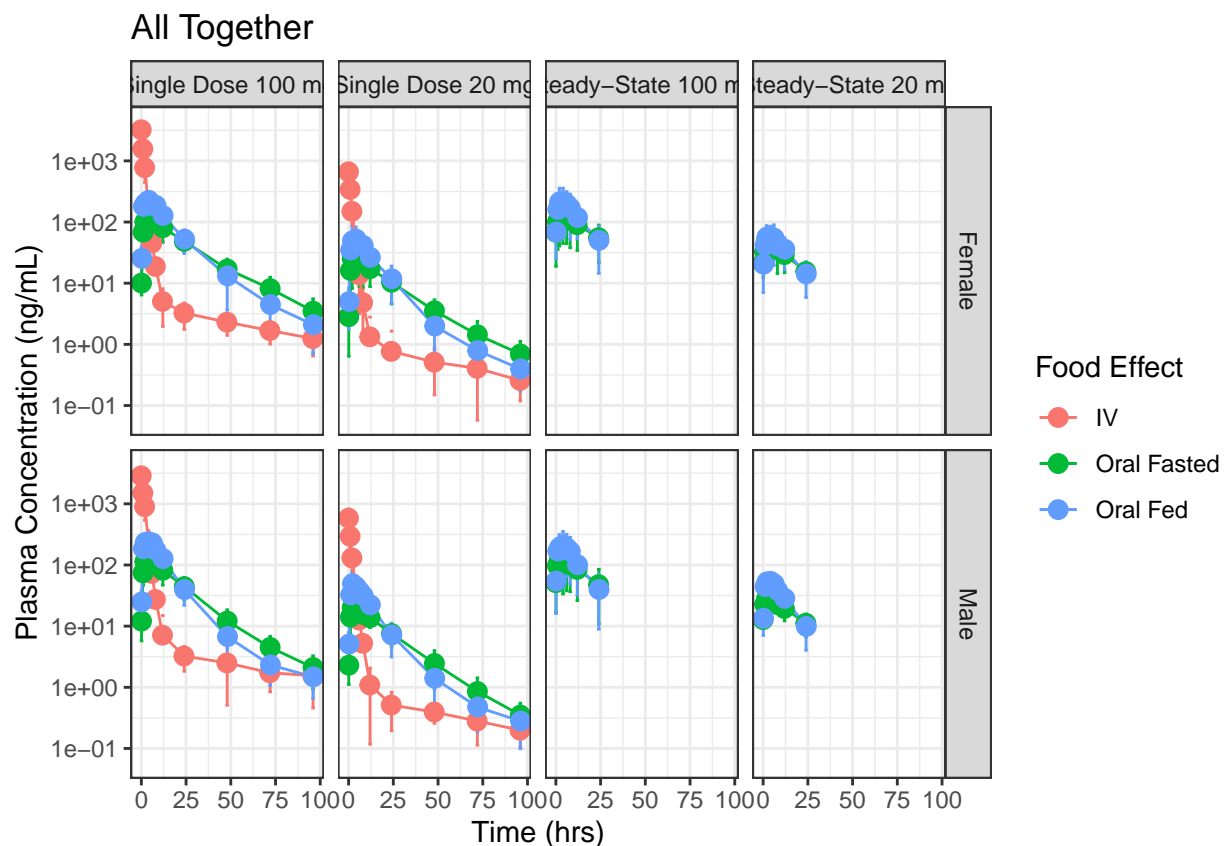
```
pd <- position_dodge(0.5)
```

```
p4<-ggplot(df.ss, aes(x=TAD, y=CONC, colour=as.factor(label3))) +
  geom_errorbar(aes(ymin=CONC-sd, ymax=CONC+sd), width=2) +
  geom_line()+
  theme_bw() +
  facet_grid(sexlabel~label2)+
  geom_point(position=pd, size=3)+
  scale_y_log10()+
  xlab("Time (hrs)") + ggtitle("All Together")+
  ylab("Plasma Concentration (ng/mL)") +
  scale_colour_discrete(name="Food Effect")
```

p4

```
## Warning in self$trans$transform(x): NaNs produced
```

```
## Warning: Transformation introduced infinite values in continuous y-axis
```



We can customize the x-axis by using `facet_wrap`

```
p5<-ggplot(df.ss, aes(x=TAD, y=CONC, colour=as.factor(label3))) +
  geom_errorbar(aes(ymin=CONC-sd, ymax=CONC+sd), width=2) +
  geom_line() +
```

```

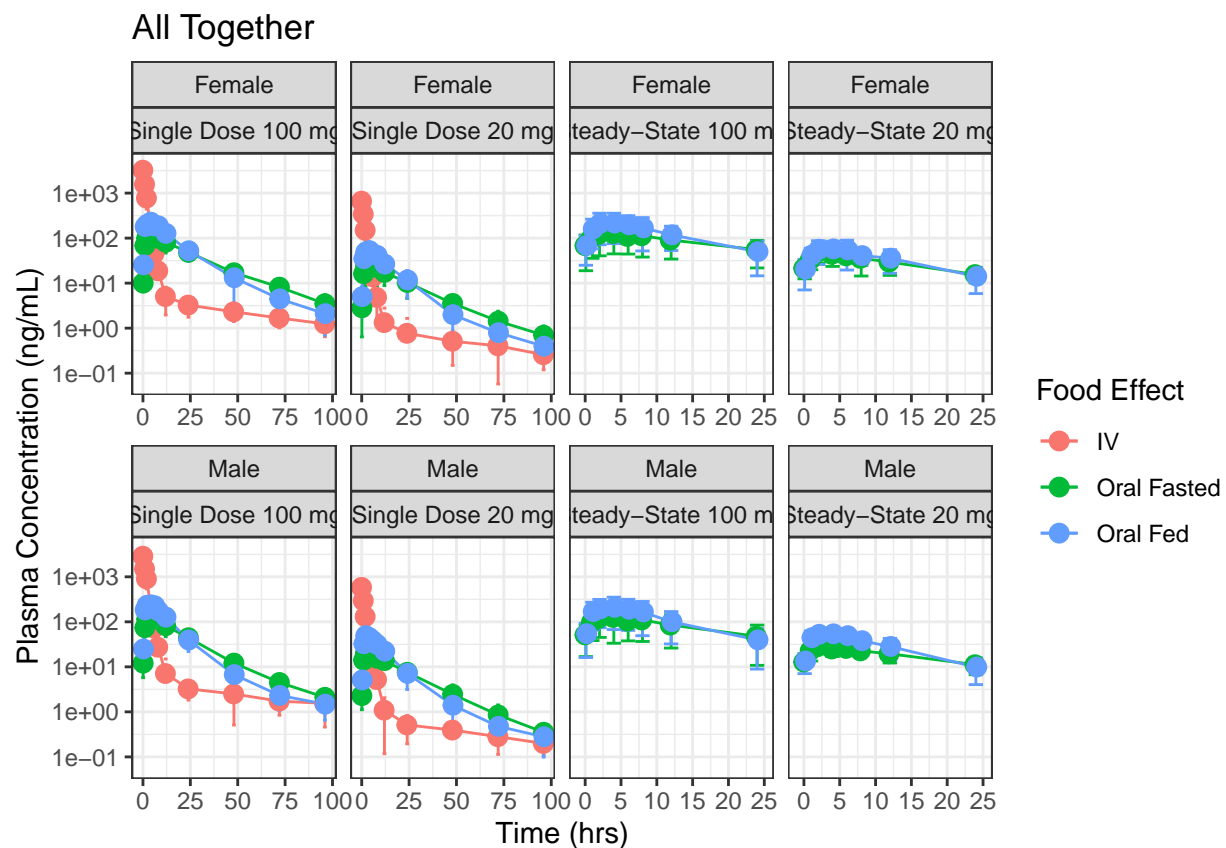
theme_bw() +
facet_wrap(sexlabel~label2,ncol=4, scales="free_x")+
geom_point(position=pd, size=3)+
scale_y_log10()+
xlab("Time (hrs)") + ggtitle("All Together")+
ylab("Plasma Concentration (ng/mL)") +
scale_colour_discrete(name="Food Effect")

```

p5

```
## Warning in self$trans$transform(x): NaNs produced
```

```
## Warning: Transformation introduced infinite values in continuous y-axis
```



Alternatively, we can use `Stat_summary` to do the same:

```

p6<-ggplot(d.all, aes(x=TAD, y=EXPDV, colour=as.factor(label3),group=as.factor(label3))) +
  theme_bw() +
  stat_summary(fun=mean,
              fun.min=function(x) mean(x) - sd(x),
              fun.max=function(x) mean(x) + sd(x),
              geom="pointrange") +
  stat_summary(fun = mean,
              geom = "line") +

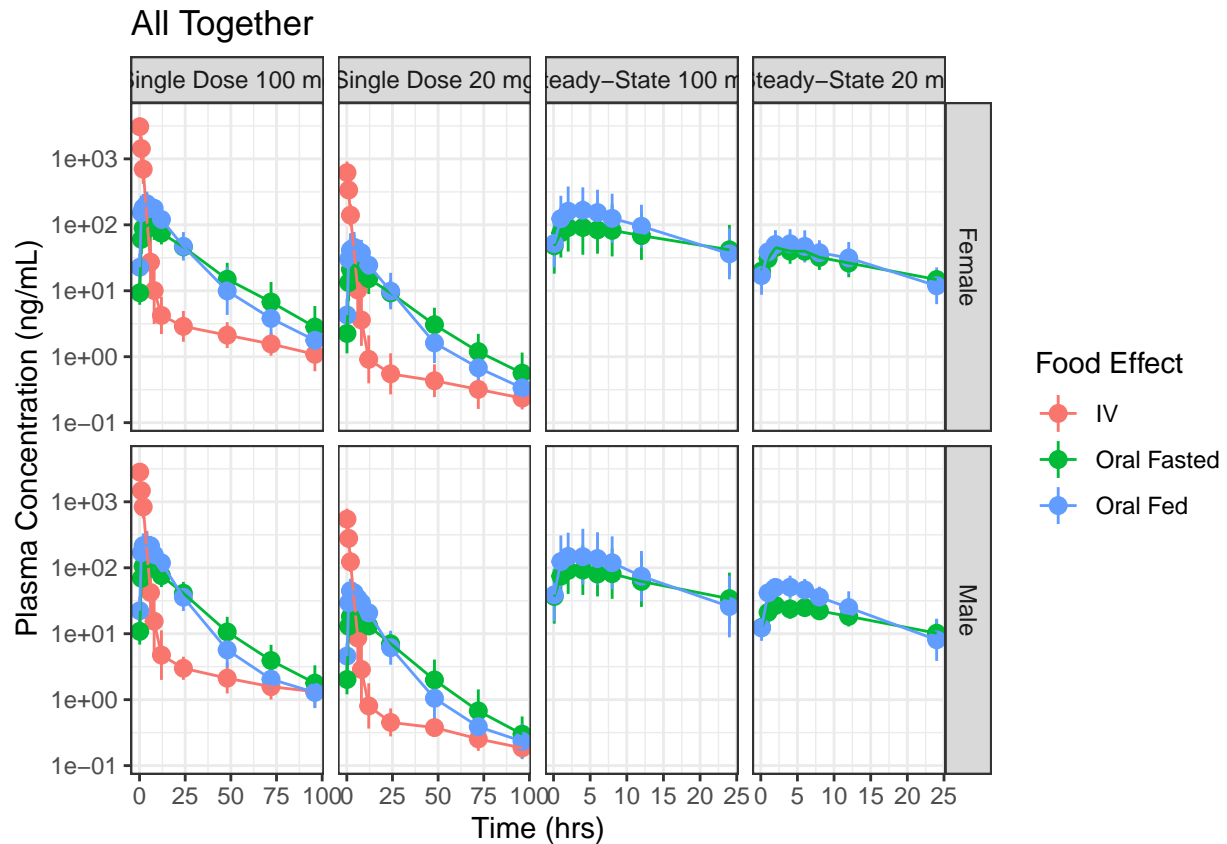
```

```

scale_y_log10()+
facet_grid(sexlabel~label2,scales="free_x")+
xlab("Time (hrs)") + ggtitle("All Together")+
ylab("Plasma Concentration (ng/mL)") +
scale_colour_discrete(name="Food Effect")

```

p6



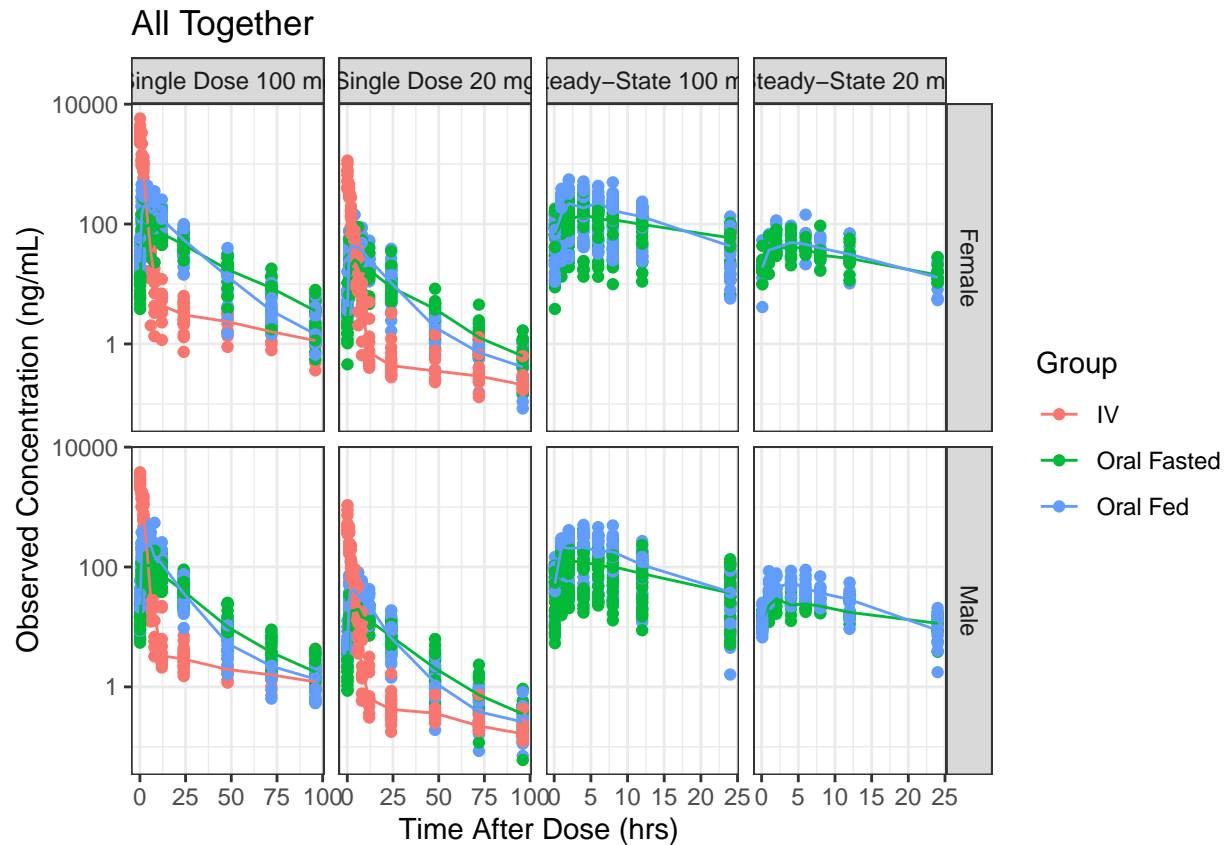
We can present the median line and observations:

```

p7<-ggplot(d.all,aes(x=TAD, y=EXPDV, colour=as.factor(label3))) +
  geom_point()+
  stat_summary(fun=median,
              geom="line") +
facet_grid(sexlabel~label2,scales="free_x")+
theme_bw() +
xlab("Time After Dose (hrs)") +
ylab("Observed Concentration (ng/mL)") +
scale_y_log10()+
ggtitle("All Together")+
scale_colour_discrete(name="Group", # Legend label, use darker colors
                      labels=c("IV", "Oral Fasted", "Oral Fed"))

```

p7



Finally, you can save your plot:

```
ggsave("20singledose.png", width=8, height=8)
```

```
mrggsave(list(p,p2,p3,p4,p5,p6,p7), tag = "all_plots",width = 7, height = 7)
```

```
## Warning in self$trans$transform(x): NaNs produced
```

```
## Warning: Transformation introduced infinite values in continuous y-axis
```

```
## Warning in self$trans$transform(x): NaNs produced
```

```
## Warning: Transformation introduced infinite values in continuous y-axis
```

```
#Advanced materials
```

```
#####individual plots#####
```

```
#Creating individual plots and saving them each page as a png
```

```
Create a Plot per Individual and Food Status
```

```
p1 <- ggplot(d.all, aes(x=TIME, y=EXPDV)) + geom_point(colour="blue", shape=1) +
  geom_line(aes(x=TIME, y= EXPDV), colour="black", size=0.5) +
  facet_wrap(~ID+label3, scales="free")+
```

```

scale_y_log10()+
ylab("Concentration (ng/mL)") + xlab("Time (hr)") + theme_bw() +
theme(strip.text = element_text(size=8))

# list out plots into grob
templots <- dply(d.all, "ID", `~%+~` , e1 = p1)

# multiple page arrangement for grobs
myplots <- marrangeGrob(templots, nrow=3, ncol=3)

ggsave("ind%02d.png", myplots, width=12, height=12)

mrggsave(myplots, tag = "individuals", width = 7, height = 7, onefile=FALSE, dev=c("pdf", "png"))

```

#Box Plot

First we create a dataset with one observation per Individual for single and multiple dose, fed and fasted. Then we create the boxplot using `geom_box`

```

one<-d.all%>%
  group_by(label2, label3, ID)%>%
  dplyr::summarise(uCMAX=max(CMAX))%>%ungroup()

```

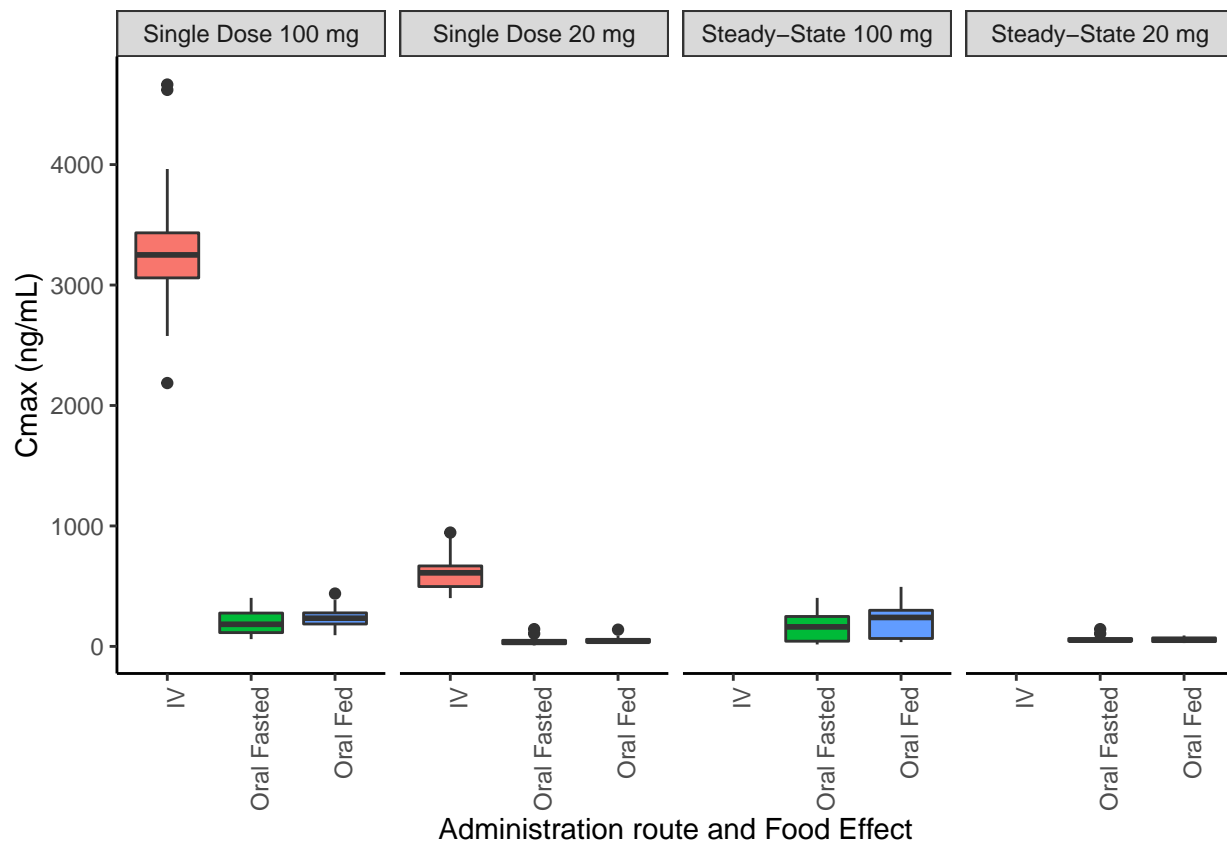
'summarise()' has grouped output by 'label2', 'label3'. You can override using
the '.groups' argument.

```

cmax.bp<-ggplot(one, aes(x=label3, y=uCMAX, fill=label3)) +
  geom_boxplot() +
  theme_bw() +
  theme(panel.border = element_blank(), panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        axis.line = element_line(colour = "black")) +
  theme(legend.position = "none") +
  facet_grid(~label2) +
  xlab("Administration route and Food Effect") +
  ylab("Cmax (ng/mL)") + theme(axis.text.x = element_text(angle = 90, hjust = 1))

```

```
cmax.bp
```

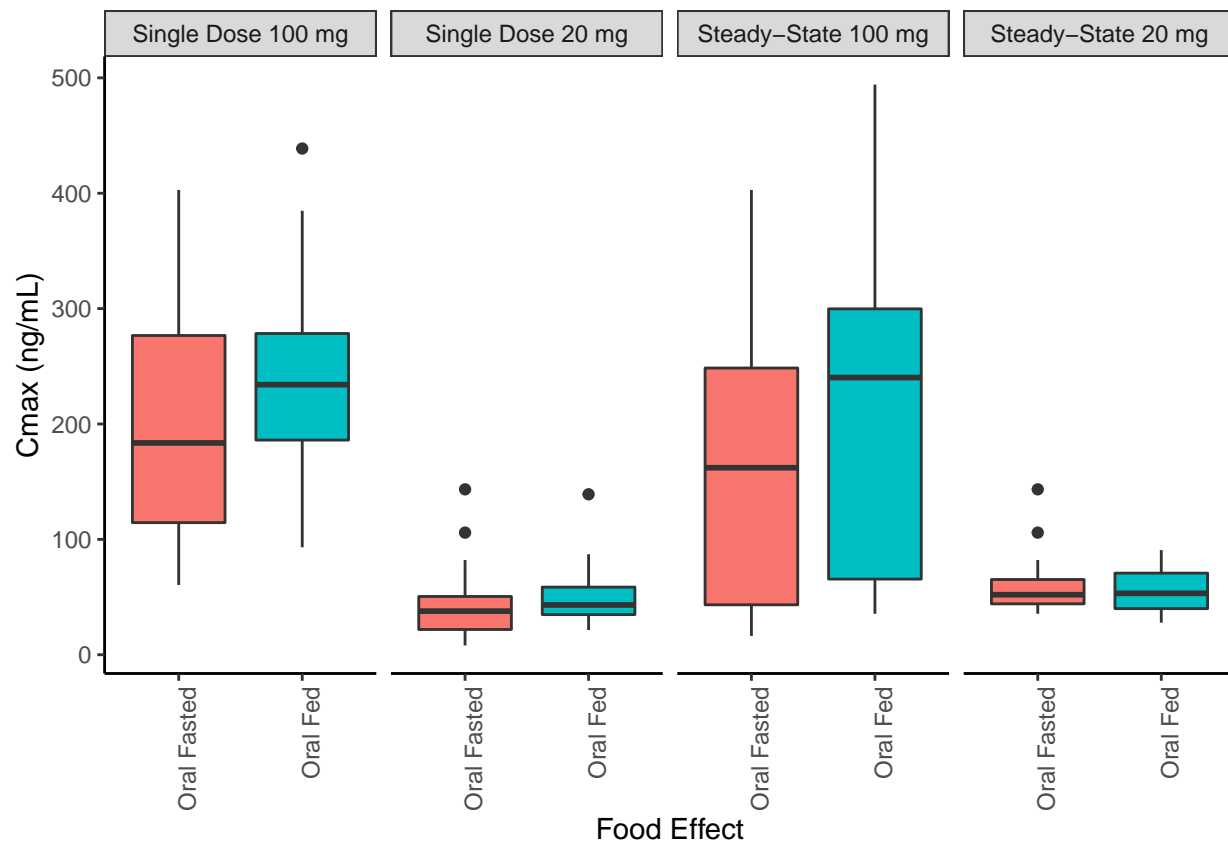


```

cmax.bp2<-ggplot(one%>%filter(label3!="IV"),aes(x=label3, y=uCMAX,fill=label3))+
  geom_boxplot()+
  theme_bw() +
  theme(panel.border = element_blank(), panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        axis.line = element_line(colour = "black"))+
  theme(legend.position = "none")+
  facet_grid(~label2)+
  xlab("Food Effect")+
  ylab("Cmax (ng/mL)") +theme(axis.text.x = element_text(angle = 90, hjust = 1))

```

cmax.bp2



```
mrsggsave(list(cmax.bp,cmax.bp2), tag = "boxplots",width = 7, height = 7)
```